Amendments to the Claims

1.-20. (Cancelled)

21. (Currently Amended) A method comprising:

for each binary file out of a set of binary files:

determining dependency information about the binary file[[s]] comprising entry and exit points for the binary file[[s]];

recording indications of the entry and exit points in a record for the binary file;

propagating dependency information about the set of binary files to determine subsystem dependency information for one or more subsystems which contain binary files out of the set of binary files by:

generating pairs of entry and exit points for binary files which are dependent on one another; and

determining subsystem dependency information comprising entry and exit points for [[a]] the one or more subsystems ~~of a system~~ based on the generated pairs of entry and exit points for the set of binary files;

propagating the subsystem dependency information to determine system dependency information by:

generating pairs of entry and exit points for subsystems in the system; and

determining system dependency information comprising entry and exit points for the system based on the generated pairs of entry and exit points for subsystems in the

system;

marking changed logical abstractions in one or more changed binary files;

marking unchanged logical abstractions which are in one or more unchanged binary files, the unchanged logical abstractions being dependent on marked changed logical abstractions in ~~other subsystems~~ the one or more changed binary files;

comparing test coverage to marked changed logical abstractions and to marked unchanged logical abstractions; ~~and~~

prioritizing tests based on maximum test coverage of marked changed logical abstractions and marked unchanged logical abstractions; and

performing the prioritized tests according to test priorities to produce test results.

22. (Original) The method of claim 21 wherein the test coverage comprises tests for one subsystem.

23. (Original) The method of claim 21 wherein the test coverage comprises tests for plural subsystems.

24. (Original) The method of claim 23 wherein the test coverage comprises tests for plural subsystems and maximum test coverage is considered for marked changed logical abstractions and marked unchanged logical abstractions for said plural subsystems.

25. (Currently Amended) A computer-based service comprising:

means for determining binary dependencies, comprising pairs of entry and ~~exist~~ exit points, for binary files in a defined system;

means for propagating the binary dependencies to identify ~~binaries~~ binary files dependent on ~~binaries~~ other binary files in other subsystems;

means for storing determined and propagated dependencies;

means for marking changes in changed binary files;

means for propagating marked changes using the determined and propagated dependencies to unchanged binary files; and

means for prioritizing tests which cover the unchanged binary files based on test coverage of marked changes and propagated marked changes;

means for performing prioritized tests according to test priorities to produce test results.

26. (Original) The service of claim 25 wherein the means for marking changes includes means for marking proposed changes.

27. (Cancelled)

28. (Cancelled)

29. (Currently Amended) A computer system comprising:

a processor coupled to volatile and nonvolatile memory;

binary files stored in memory;

software stored in memory comprising computer executable instructions for:

~~determining~~ recording dependency information for binary files comprising entry and exit points for the binary files;

propagating dependency information about binary files to determine subsystem dependency information comprising entry and exit points for a subsystem of a system; and

propagating subsystem dependency information to determine system dependency information comprising entry and exit points for the system;

marking logical abstractions changed from a previous version in one or more changed binary files;

propagating marked changes according to the dependency information comprising marking unchanged logical abstractions in one or more unchanged binary files dependent on marked changes in ~~other subsystems~~ the one or more changed binary files;

comparing test coverage to marked changed logical abstractions and to marked unchanged logical abstractions; and

prioritizing tests based on maximum test coverage of marked changed logical abstractions and marked unchanged logical abstractions.

30. (Original) The computer system of claim 29, wherein maximum test coverage is based on the total number of marked changed and marked unchanged logical abstractions touched by a test system wide.

31. (Currently Amended) The computer system of claim 29, wherein maximum test coverage is based on the sum of the total number of marked changed logical abstractions in a first subsystem touched by a test, and marked unchanged logical abstractions touched by the test in the first subsystem, wherein the marked unchanged logical abstractions depend on marked changed logical abstractions in binary files in other subsystems.

32.-36. (Cancelled)